Database Management Systems
Practice Exam Questions

Question 1 Answer true (T) or false (F).

____ 1. A relation must have a clustered index on its primary key.

____ 2. If R.B is a (non-null) foreign key referencing S.B, then the number of tuples in $R[A, B] \bowtie S[B, C]$ is $T(R)$ (that is, the number of tuples in R).

____ 3. Using a nested loop join, $R \bowtie S = S \bowtie R$.

____ 4. A sort-merge join is always more efficient than an index join.

____ 5. A linear hash index cannot be clusterd.

____ 6. The cost of external sorting is independent of the buffer size.

____ 7. Let $M > 5$ be the number of memory blocks available to process a join on R and S. If the size in blocks of R less than M^3 then a hash join (assuming a good hash function) can be done in two passes (one to partition and one to join).

Question 2 Query Processing

Consider the following query on the tables Emp and Dept declared below: “select Name, DeptHead from Emp, Dept where Emp.Dept = Dept.Name”. Assume a data page can hold 50 tuples. Assume a B$^+$-tree index page can hold 100 key-pointer pairs. Let $t(emp) = 1,000,000$ and $t(dept) = 500$. Assume there is at least one employee in every department. The DBMS is capable of performing nested loops, index-nested loops, and sort-merge joins. For simplicity, assume the cost of sorting Emp is 22,000 and the cost of sorting Dept is 30. In all cost computations, do not include the cost of storing the result of the join. Four buffer pages are available for doing the join.

`Emp[`<u>`Name`</u>`, Location, Manager, Salary, Dept], Dept[`<u>`Name`</u>`, Headquarters, DeptHead]`
Emp.Name primary key of Emp and Dept.Name primary key of Dept
Emp.Dept is foreign key of Dept

1. Given a clustered B$^+$-tree index on Emp.Dept and a unclustered B$^+$-tree on Dept.Name, describe the lowest cost join plan (and give its estimated I/O cost).
2. For practice, think about this question with different indices, different join methods, different statistics, and different buffer sizes.
3. An alternative question might ask about methods for performing other relational operators than join (e.g., selection, projection, duplicate elimination, union, intersection, differences, etc.)